

Graphviz

Open source graph visualization software

Uvod

- Graphviz je open source software za vizualizaciju grafova
- Vizualizacija grafova pruža način za reprezentaciju strukturnih informacija kao dijagrama apstraktnih grafova i mreža
- Programi za layoutiranje Graphviz-a primaju opis grafa u jednostavnom tekstualnom jeziku i kreiraju dijagrame koji se snimaju kao slike ili pdf dokumenti

Instalacija

- Skinuti odgovarajuću instalaciju sa zvaničnog sajta
- Na Windows-u dodati putanju do bin direktorijuma Graphviz-a na PATH varijablu
 - npr. C:\Program Files (x86)\Graphviz2.38\bin

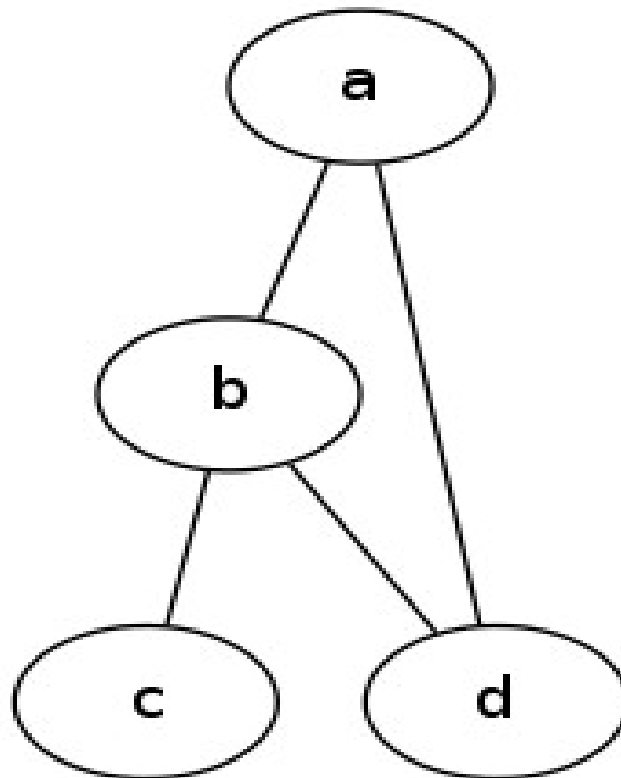
DOT jezik (1)

- Grafovi u se specificiraju upotrebom DOT jezika
- Primer jednostavnog grafa u DOT jeziku:

```
graph graphname {  
    a -- b;  
    b -- c;  
    b -- d;  
    d -- a;  
}
```

DOT jezik (2)

- U primeru je definisan neusmereni graf koji ima 4 čvora (a, b, c i d) i 4 ivice koje ih povezuju:



DOT jezik (3)

- Ako želimo da kreiramo usmereni graf, umesto ključnom rečju *graph*, opis započinjemo rečju *digraph*
- Ukoliko želimo da napravimo vezu koja polazi iz čvora *a* i završava se u čvoru *b*, zapisujemo je u obliku *a -> b*

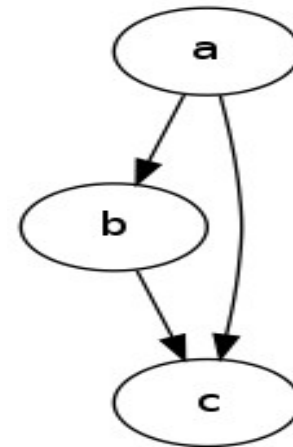
```
digraph graphname{
```

```
    a -> b;
```

```
    b -> c;
```

```
    a -> c;
```

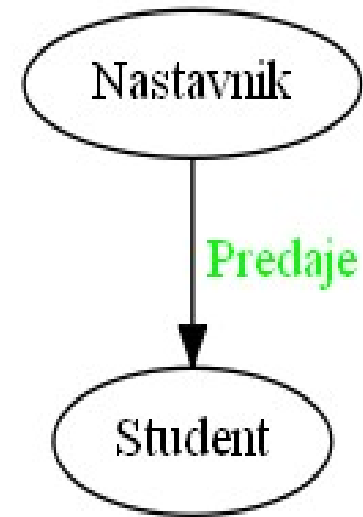
```
}
```



DOT jezik (4)

- Čvorovima i ivicama možemo dodeliti labele, kao na sledećem primeru:

```
digraph somegraph{  
  N [label = "Nastavnik"]  
  S [label = "Student"]  
  N -> S [label = " Predaje",  
    fontcolor = green]  
}
```



- Komentari se pišu kao u Javi ili C-u, nakon // ili između /* i */

DOT jezik (5)

- Na primeru se vidi i da možemo specificirati neka dodatna vizuelna obeležja, poput boje fonta
 - možemo podesiti i veličinu fonta, boju ivice i unutrašnjosti čvora, kao i njihov oblik (krug, elipsa, trougao, tačka...)
 - možemo podešavati i oblik i labelu početka i kraja veze
 - šta je sve podržano od oblika za čvor može se naći na sledećem [linku](#)
 - a za ivicu na sledećem [linku](#)

DOT jezik (6)

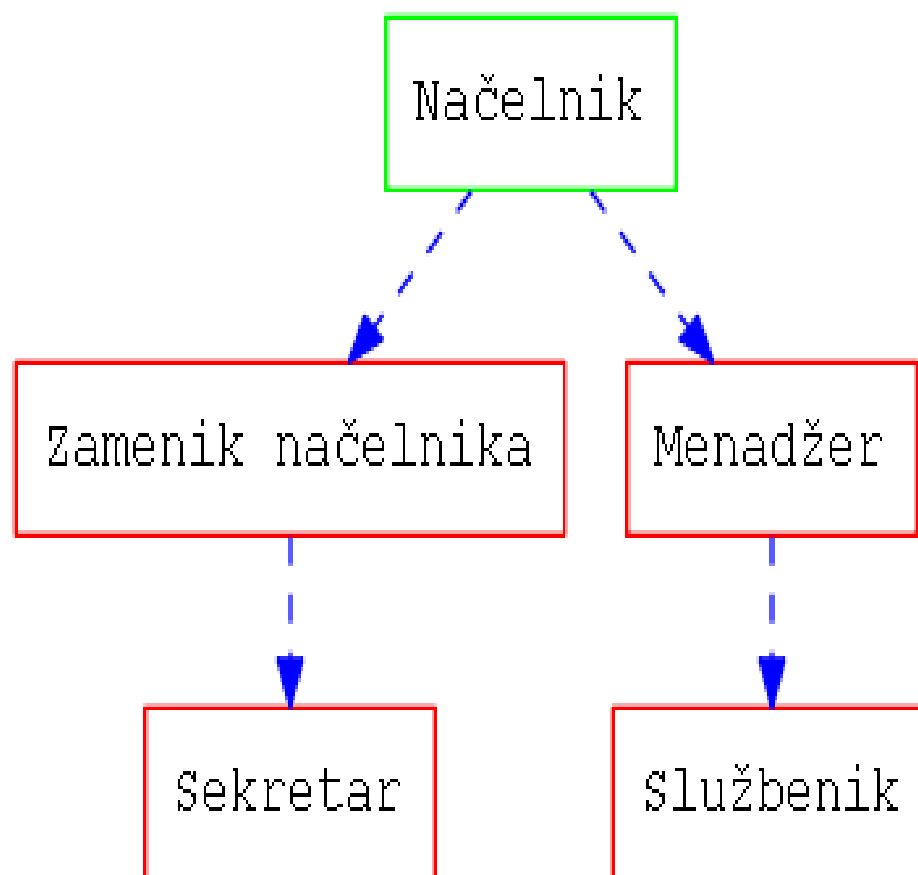
- Kako se ne bi za svaki čvor i ivicu pojedinačno podešavali željeni atributi, mogu se zadati i globalno za sve čvove i ivice
- Na primer, `node [color = Red]` specificira boju svih čvorova
- Specificirana obeležja se mogu podesiti i na nivou pojedinačnih čvorova, koja onda dobijaju prioritet u odnosu na globalna
- Takođe, možemo nanizati više veza (`A->B->C` je isto što i `A->B` i `B->C`)

DOT jezik (7)

```
digraph hierarchy{  
    node [color = Red,fontname=Courier,shape=box]  
    edge [color = Blue, style=dashed]  
    nacelnik[label = "Načelnik", color = Green]  
    zamenik[label = "Zamenik načelnika"]  
    manager[label = "Menadžer"]  
    sluzbenik[label = "Službenik"]  
    sekretar[label = "Sekretar"]  
    nacelnik -> manager -> sluzbenik  
    nacelnik -> zamenik -> sekretar  
}
```

DOT jezik (8)

- Rezultujući dijagram:



DOT jezik (9)

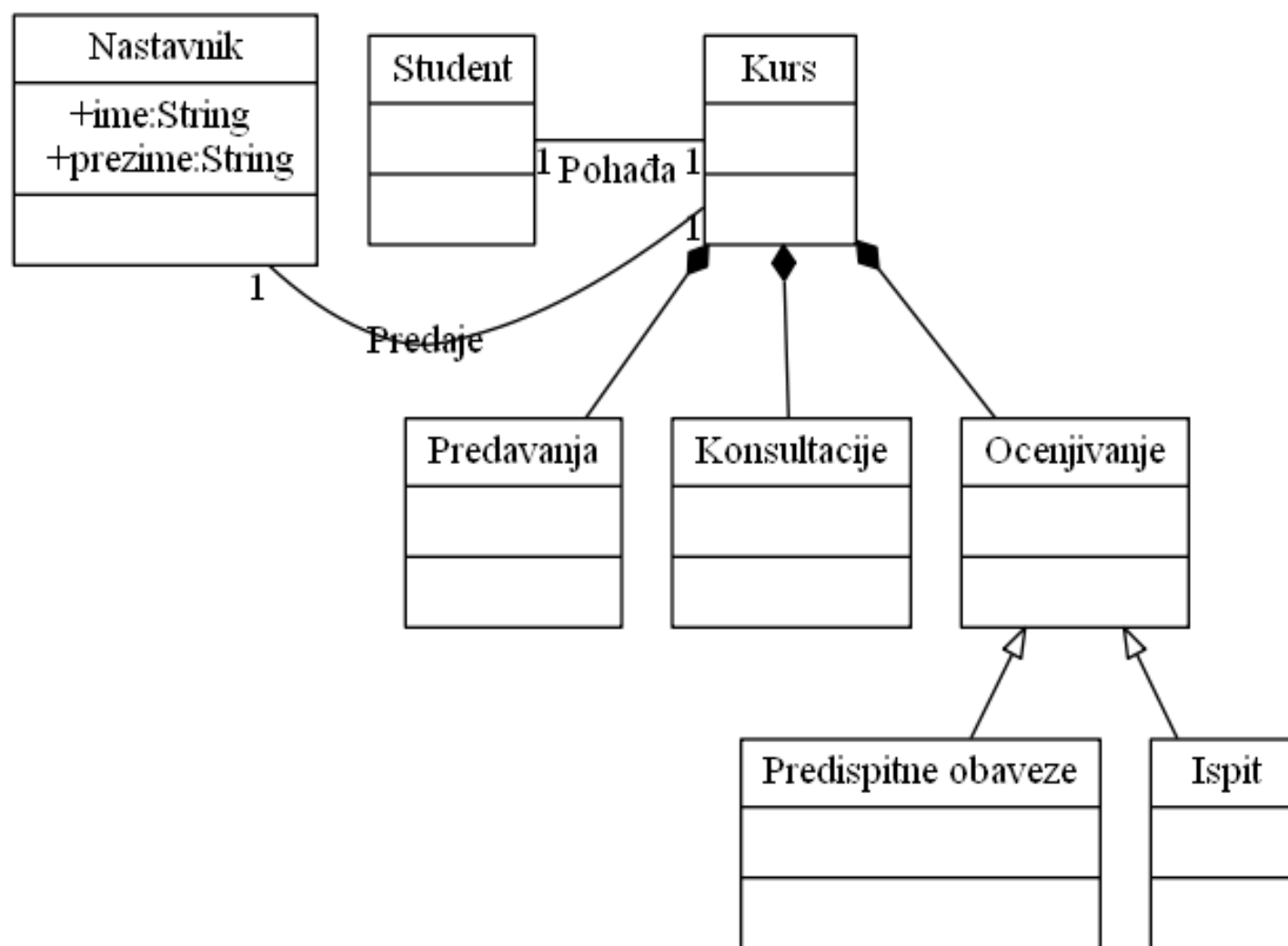
- Objekti model:

```
digraph objectmodel{  
    node[shape=record];  
    rankdir="BT";  
    teacher [label = "{Nastavnik| +ime:String \n +prezime:String|\n}"];  
    course [label = "{Kurs|\n|\n}"];  
    student [label = "{Student|\n|\n}"];  
    lesson [label = "{Predavanja|\n|\n}"];  
    tutorial [label = "{Konsultacije|\n |\n}"];  
    assessment[label = "{Ocenjivanje|\n|\n}"];  
    coursework [label = "{Predispitne obaveze|\n|\n}"];  
    exam [label = "{Ispit|\n|\n }"];
```

DOT jezik (10)

```
{rank=same; teacher course student} //označava da su na istom nivou  
teacher->course [label="Predaje",arrowhead="none",  
arrowtail="normal",headlabel="1",taillabel="1"];  
student->course [label="Pohađa", arrowhead="none",  
arrowtail="normal",headlabel="1",taillabel="1"];  
lesson->course [arrowhead="diamond",arrowtail="normal"];  
tutorial->course [arrowhead="diamond",arrowtail="normal"];  
assessment->course [arrowhead="diamond",arrowtail="normal"];  
coursework->assessment[arrowhead="onormal"];  
exam->assessment[arrowhead="onormal"];  
}
```

DOT jezik (11)



DOT jezik (12)

- Atribut *rankdir* upotrebljen u primeru postavlja usmerenje grafa
 - na primer, vrednost “LR” označava sa leva prema desno, a “BT” od dole prema gore
 - po default-u je od gore prema dole
- Pregled svih atributa grafova, čvorova i ivica se može naći na [linku](#)

Kreiranje grafova iz aplikacije

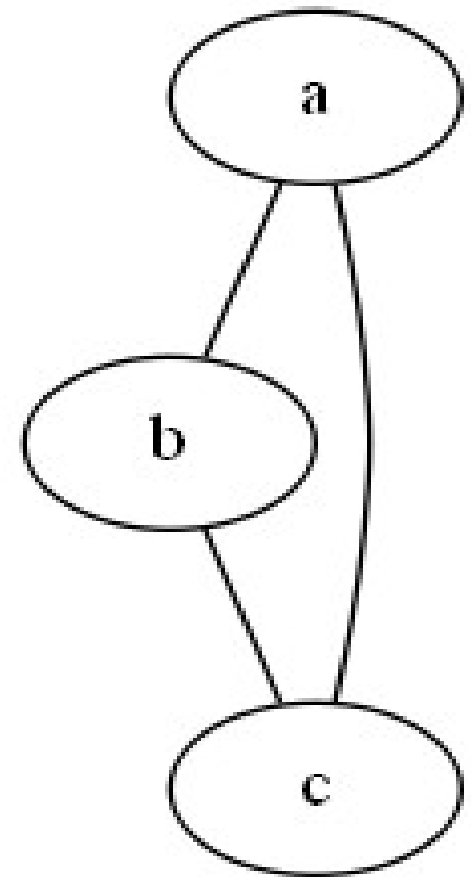
- Fajlovi u koje se snima opis grafova u dot jeziku treba da imaju ekstenziju .dot
- Iz aplikacije ih možemo kreirati ručnim zadavanjem koda kojim treba popuniti ove fajlove (pogledati recimo export modul textX-a)
- Možemo i koristiti neki od Python modula koji pružaju interfejs ka dot jeziku
 - pydot, graphviz, PyGraphviz
- Najčešće se koristi pydot

Pydot

- Python interfejs ka Graphviz-u
- Modul pruža mogućnost kreiranja, modifikacije i procesiranja grafova u Graphviz-ovom dot jeziku
- Instalacija:
 - prvo instalirati pyparsing - za parsiranje dot fajlova (pip install pyparsing) i Graphviz
 - ako se koristi Python 2 instalirati preko pip-a: pip install pydot
 - ako se koristi Python 3 skinuti kod sa [linka](#) i pokrenuti setup.py

Jednostavni graf kreiran preko pydot-a

```
from pydot import Node, Edge, Dot  
  
#graph_type - graph ili digraph  
  
graph = Dot(graph_type = "graph")  
node1 = Node("a")  
node2 = Node("b")  
node3 = Node("c");  
edge1 = Edge(node1, node2)  
edge2 = Edge(node2, node3)  
edge3 = Edge(node1, node3)  
graph.add_edge(edge1)  
graph.add_edge(edge2)  
graph.add_edge(edge3)  
graph.write_png("primer1_graph.png")
```



Podešavanje atributa (1)

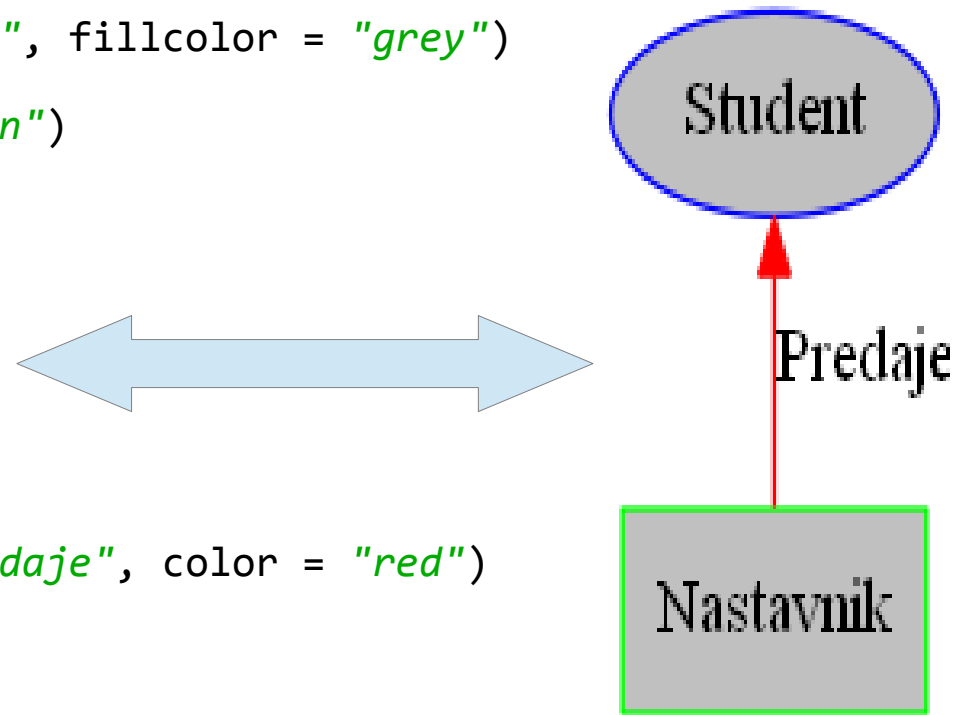
- I preko pydot-a možemo podesiti attribute grafa, čvorova i ivica, bilo prilikom kreiranja odgovarajućih objekata, bilo nakdandno preko set metode
- Recimo, `Node("A", color = "green")` - kreira čvor sa ivicom zelene boje
- Ili, preko set metode: `node.set("color", "green")`
- Ili na nivou celog grafa:
`graph.set_node_defaults(color = "green")`

Podešavanje atributa (2)

- Slično, možemo podesiti i osobine ivica - pri konstrukciji objekta, preko set metode ili putem set_edge_defaults metode grafa
- Spisak svih podržanih atributa možemo videti pregledanjem koda pydot modula, [link](#)
- Primećuje se da se radi o istim nazivima kao u dot jeziku

Malo kompleksniji primer

```
graph = Dot(graph_type = "digraph", rankdir = "BT")
graph.set_node_defaults(style = "filled", fillcolor = "grey")
node1 = Node("Nastavnik", color = "green")
node2 = Node("Student", color = "blue")
node1.set("shape", "record")
graph.add_node(node1)
graph.add_node(node2)
edge1 = Edge(node1, node2, label = "Predaje", color = "red")
graph.add_edge(edge1)
graph.write_png("primer2_graph.png")
```



Podešavanje rank obeležja

- Da bi se moglo specificirati da se nekoliko čvorova nalaze na istoj visini (ili širini ao se radi o LR ili RL usmerenju) moramo kreirati podgraf, te u njega dodati željene čvorove i podesiti rank:

```
S = Subgraph(rank= 'same')
```

```
S.add_node(node1)
```

```
S.add_node(node2)
```

```
S.add_node(node3)
```

```
graph.add_subgraph(S)
```

Snimanje u dot fajl

- Ako želimo samo snimiti graf u dot fajl, to možemo postići snimanje ispisa njegove `to_string` metode u fajl
- Ako je potrebno snimiti u utf-8 formatu, možemo koristiti `codecs` modul

```
import codecs
```

```
with codecs.open("primer.dot", "w", "utf-8") as file:  
    file.write(graph.to_string())
```

Zadatak

- Napraviti objektni model iz primera DOT jezika preko pydot-a